

Some Issues in Numerical Simulation of Nonlinear Structural Response

H. D. Hibbitt .
Hibbitt, Karlsson and Sorensen
Providence, R.I.

1. Introduction

For some considerable time the author has been involved in the development of commercial finite element software, and this paper is written from that perspective. One of the remarkable features of the finite element method is its generality, and there is no better reflection of this than the observation that our company and several of our competitors each market one product only—a single, “general purpose” finite element code (our particular code is called Abaqus). These codes provide practical tools that are used in an astonishingly wide range of engineering applications that include critical aspects of the safety evaluation of nuclear power plants or of heavily loaded offshore structures in the hostile environments of the North Sea or the Arctic, major design activities associated with the development of airframes for high strength and minimum weight, thermal analysis of electronic components, and the design of sports equipment. For various reasons, the code that my company develops and markets is generally used in the area of more advanced applications. These applications almost always involve nonlinear effects. There is little doubt in my mind that the need for such analysis will continue to grow—it is very easy to identify problems which should be reachable in the next generation or two of computers and software and which have substantial economic importance.

The development, maintenance and support of production software involves many activities, but—at least in the more advanced application areas where we try to contribute—the effectiveness of our product depends critically on the quality of the mechanics and mechanics related algorithms that we implement. It is generally true that the end users are not sophisticated with respect to what is now being called “computational mechanics.” They have other interests and motivations, not the least of which is the need to complete work successfully and to a schedule. Thus, “algorithmic robustness” is of primary concern to us: we should choose those methods that we believe will maximize reliability with minimal understanding on the part of the user. Computational efficiency is important because there are always limited resources, and hence problems that we would like to do but which are too time consuming or costly. In reality, we compromise: for example, we

PRECEDING PAGE BLANK NOT FILMED

knowingly commit what Strang and Fix (1973) call "variational crimes" because we get away with them often enough for it to be worthwhile. But robustness implies a need for thorough understanding of the algorithms: we should at least know where the limitations of an algorithm are likely to be. This is not a simple task: for example, we still do not have practical ways of assessing local error, even in a linear numerical solution.

It is easy to identify important practical problems that, presently, are computationally rather difficult, but which will become relatively routine in the not too distant future. Two that are taking up much of our time just now are the problem of simulating vehicle (especially automobile) crashes, and the simulation of rather complicated contact situations, such as the analysis of threaded connectors in drill pipe or casing which is subjected to very large axial forces, causing possible thread jump and quite substantial strains in the pipe. Both problems are modeled today on current generation computers (the Cray-1 and X-MP) with Abaqus and other codes. They challenge the limits of the algorithms in our code, and are computationally intensive—typical run times are several hours per case. The observations made in this paper are based on our experience to date with such problems.

Large scale general purpose codes have a rather long life span: two codes that are widely used at this time (Ansys and Nastran) were begun in the mid 1960's. Thus, in designing such a code, it is important to try to anticipate what sort of computers will be in general use for such applications in 10–15 years. Based on past history, this is a difficult extrapolation—computers are still developing at a rapid rate. We can make some guesses. For example, at Cray Research's Science and Engineering Symposium held in Minneapolis in 1985 Seymore Cray discussed the specification of the Cray-3, which he expects will be available in three years. The most relevant parts of that specification from our point of view are the size of the high-speed, directly addressable, memory (10^9 words), and the machine's parallelism. Consider a problem of order 50000 unknowns: many important cases of the types mentioned above are of this size or smaller. The rms half-bandwidth in such a case would be about 5000, so that the assembled symmetric part of the Jacobian (stiffness) matrix will occupy 250×10^6 words—a quarter of the available memory; the complete matrix will occupy half the memory. (More and more problems are likely to need the full Jacobian matrix in the context of Newton methods, because of the use of more realistic constitutive models, such as damage mechanics models for brittle or composite materials, which have a non-symmetric Jacobian). Element matrices, state variables for constitutive calculations, etc. are unlikely to occupy more than 5×10^6 words in such a case. Thus, it would seem that, for such applications, we need not worry about "I/O" problems—we can assume the model can run "in-core." This means that, for our purposes, the practical computational limitation will be the time taken to do the arithmetic. Cray Research and others seem to be moving rapidly toward parallelism to provide arithmetic speed. This should fit well with that part of our finite element codes that perform element and constitutive calculations: there we process a large number of such calculation points (perhaps 4000 elements, 30000 integration points in the 50000 degree-of-freedom case) so that, provided the code is designed to allow inner loops to spread over the processors, load balancing should not be much of a problem. It is interesting to note that, in this part of the code, typical vectors are of order 10–100, so that any vector processor with a long start-up time (like the Cyber 205) is not very suitable. The solution of the linear equations

is not such a natural fit into parallel architecture, but the problem is such an important one that it is likely that very effective modules will become available (such as Floating Point Systems now provide for their processors). Since we in fact are solving a nonlinear system, it may be that non-Newton methods, not requiring the resolution of a Jacobian matrix, may be preferable in such an environment. I assume that load balancing is the critical issue in a multi-processor machine, and that the most effective approach to that issue is at the inner loop level, rather than the macro (finite element) level.

The remainder of the paper discusses three principal areas. First, the paper discusses our experiences with the approaches we use to the two initial value problems of primary interest here—static and dynamic nonlinear response of structures. It then has a brief section on shell modeling. Then it discusses our current approach to the constitutive integration problem, in the context of conventional plasticity models. Finally, the paper lists some areas where we hope that research work will provide us with new methods, or with improvements to our present approaches.

2. Experience with initial value problems

2.1 Structural dynamics

The nonlinear dynamic problems that we most often see involve globally nonlinear response—most of the structure yields and undergoes finite rotations and, possibly, finite strains. However, locally nonlinear dynamic problems are not uncommon: in fact they represent a sufficiently important application area that we have found it worthwhile to provide methods to model them with some efficiency. A good example of such a case is the flow induced vibration of a steam generator tube in a nuclear plant. The tube rattles in its support plates due to vibration excited by instabilities in the coolant flow, and this intermittent contact causes wear and, eventually, may lead to leakage. The tube itself may be “sprung”—that is, the initial stress in the tube may be large enough that it forms a significant contribution to the tube’s stiffness—but the vibration amplitude is not large enough that the stiffness of the tube changes significantly during the motion. The nonlinearities are therefore confined to the support interaction, where they involve intermittent, “chattering” contact and friction. Our approach to such problems has been through conventional Guyan reduction, retaining the transverse displacements at the support points and enough other degrees of freedom in the tube to model its response accurately in the frequency range of interest (up to about 150 Hz in this case). The Guyan method has two drawbacks—it is based on guesswork, and the reduced model may be much bigger than necessary. Since the majority of the applications that have come to our attention involve relatively small models (typically less than 1000 degrees of freedom, with 30–80 degrees of freedom directly involved in local nonlinearities), these limitations are not very critical: it is not much effort to extract the modes of interest on the full model and then to experiment with reduced models until a reasonably small one is found that provides adequate matching.

Nonlinear dynamic problems are integrated explicitly or implicitly. The explicit approach brings with it several advantages because it is always used with a lumped mass

matrix and so no equations need be solved: this leads to great simplification in the computer code. But conditional stability is a serious limitation, especially in structural models where the thickness of the shell is usually the determining factor with respect to the time step. Many of the problems that we see are "event and response" cases, in which an initial input of energy is mostly dissipated as inelastic response in the structure. In these cases the response usually damps out fairly quickly into a pattern of plastic hinges, until some later disturbance—perhaps a secondary impact, or a sudden effect associated with geometry changes—causes this pattern to undergo a redistribution. Indeed, this phenomenon of plastic hinge formation is so much a part of such problems that it has given rise to the successful mode form solution approach of P. S. Symonds, J. B. Martin, and others (Symonds, 1967). It is difficult to accept that a conditionally stable integration method which can never expand the time step beyond some fraction of the shortest period exhibited by the model provides the optimal approach for such an application. My own experience with explicit methods is limited, so that I do not know how one deals with constraints (such as arise in mixed element formulations) within these methods. But if the methods are only useful when we do not solve equations, it seems that the generality of their application is limited.

Implicit methods are chosen for their numerical stability. Stability is usually discussed in the context of linear systems, and I am not sure of what stability proofs exist for the nonlinear problems we are trying to model. My own practical experience has been that numerical stability of the operator has never been a limitation: except for accuracy considerations, the time step is most usually limited by our ability to solve the equations. I will return to the equation solving problem later, because it is, in our experience, a key issue. Implicit methods have the great advantage of generality—the time step can be chosen for modeling reasons. And since we have to solve equations anyway, we are at liberty to introduce whatever additional equations we care to, including, for example, Lagrange multipliers to impose constraints. In Abaqus we have for some time been using the Hilber-Hughes-Taylor (1978) operator (an extension of the trapezoidal rule that provides the ability to introduce some numerical damping) with a simple "automatic" time step selection scheme, and this approach has been of great practical benefit. Again, consider the car crashworthiness problem. A front end collision test case for a car design usually involves about 10–15 milliseconds of response. With the shell elements in Abaqus and a mesh that is adequate to model the response usefully, the stability limit of the central difference operator is in the microsecond range. Using the implicit method, the analysis is usually completed in 150–300 time increments, which range from a few microseconds just after a major impact up to a significant fraction of a millisecond during the fixed plastic hinge régime. The utility of the method then depends on whether we can solve the nonlinear equations 150–300 times in an acceptable amount of computer time.

2.2 Statics

Smoothly nonlinear static problems are not uncommon, but it seems that many important applications involve abrupt changes in the response, which is often unstable as well. The car crash problem again serves to provide an example. During a front end crash, most conventional designs of front rails in cars are responding unstably during about 80% of

their usable deformation, and the switch into this collapsing response occurs quite sharply, as a buckling of the structure (which is a shell), after it has undergone considerable plastic deformation. Most shell buckling cases involve relatively thin shells, so that the equilibrium equations are not well conditioned. Elastic-plastic buckling often involves sudden “localization” of the plastic deformation into narrow regions, while major parts of the structure, which were yielding in the pre-collapse phase, unload elastically. Such problems make demands on the solution strategy—it must be able to detect, and to switch into, the alternate equilibrium path, and the instability of the collapse phase of the response must be handled. For the latter purpose we have found that our version of a “Riks” algorithm (Riks, 1979) has been most valuable. We use it with an “automatic” incrementation scheme, and find that it can often march right through into the collapse as far as we want to go, without too much stuttering. The most common complaint that we receive is that the solution sometimes turns back on itself. When this occurs it is at critical points where, presumably, the equilibrium path has very high curvature. For the present we ascribe this to a weakness in our implementation only: the method finds an equilibrium path, but it is not the path of interest. We have nothing in our code that can detect the possibility of alternative equilibrium paths: for example, we know we cannot obtain sensible results for a round bar in compression (Euler buckling) or tension (necking), without knowing that the switch may occur and seeding the problem definition with a suitable imperfection. A robust algorithm would not need this.

Rate form constitutive models (such as conventional plasticity or visco-plasticity theories) still must be integrated, even though the overall problem is quasi-static. There seems to be relatively little motivation for choosing explicit methods in general for this aspect of the problem, although there are some particular cases where an explicit approach makes sense: for example, many high temperature creep problems associated with metal structures can be treated efficiently with explicit integration of the creep model, because in these cases the response times of interest are usually not very long compared to characteristic relaxation times for the material subjected to the stresses that arise in the structure—otherwise the design would be unacceptable anyway. In Abaqus, except for this particular case, we use implicit integration for rate plasticity models, as discussed below.

3. Equation solving

Almost all of the procedures we use in Abaqus are based on a “full” Newton method. We have tried a few alternatives (modified Newton and quasi-Newton methods), but so many problems of interest are not very well conditioned and exhibit such knotty response that we have not looked at many of our standard test cases before we have rejected the alternatives that we have tried, and returned to the quadratic convergence of the full Newton method. However, our work is hardly rigorous or complete.

A significant part of the usage of our code involves systems for which the Jacobian matrix of the Newton method is not symmetric. Examples are the non-associated flow plasticity models that are often used in soil mechanics applications, and loading cases with “follower forces”, like Morison drag on offshore pipes and risers. In these cases we usually form and solve the non-symmetric system. There are some problems where we offer the

possibility of approximating the Jacobian with its symmetric part because our experience has suggested that many typical cases of that class are analysed with less computational expense in that way. An example is mentioned later in this paper in connection with integration of plasticity models. But we always provide the possibility of invoking the non-symmetric capability if needed, typically by restarting the simulation at some point, because it often happens that the symmetric approximation becomes less effective as the solution develops.

Newton's method introduces difficulties of its own. The most expensive is the need to form and solve a system of linear equations at each iteration, and the most awkward is the need to define the Jacobian matrix. The algebraic manipulations involved in defining this matrix can be formidable—and there are obviously cases when it cannot be defined, except perhaps numerically. For example, a recent paper on numerical methods in plasticity (Simo and Ortiz, 1984) contains the remark:

“In general, however, the task of evaluating the consistent tangent moduli in closed form may prove exceedingly laborious. It would appear, therefore, that a general purpose implementation of the physically more compelling algorithm ... may require the use of quasi-Newton or secant-Newton methods ...”

This difficulty should not be underestimated.

The straightforward approach we have been using for conventional plasticity models is discussed in Section 5. **One further comment concerning that method is appropriate here:** even in cases when the rate plasticity model exhibits the usual symmetry property obtained from assuming associated flow and a smooth yield surface, the exact Jacobian of the integrated model—using the integration operator that we have chosen—is not always symmetric.

The full Newton method is expensive computationally. We can provide some numbers to quantify this on computers that are available today. Abaqus is a general purpose code and is used on many different computer systems, presently ranging from the Apollo 300 to the Cray X-MP. The code is not particularly “tuned” to any system, and therefore should be representative of typical straightforward finite element codes (we know from benchmarks that this is generally true). We use some standard benchmark problems to estimate performance of our code on various computers: experience has shown that these benchmarks are reliable. One of these is a shell model, with 1000 eight-node elements. It has about 17000 degrees of freedom and an rms wavefront of 400 degrees of freedom. Such a mesh would be adequate for the typical front-end collision analysis that I used as an example above. On a CrayX-MP this problem takes 48 cp seconds per iteration: 60% of that time is in the linear equation solver, and most of the rest is associated with element and constitutive calculations. A typical dynamic analysis of a front end collision needs 150–300 time increments with the implicit operator that we use, and between three and four iterations per increment with the “almost full Newton” implementation of the shell in Abaqus (the initial stress terms associated with bending are not defined exactly in the Jacobian—there is no basic problem, we have simply not completed some lengthy manipulations. I do not think the terms we omit are very significant). Thus, such a crash

simulation can be expected to pass through the basic loop about 500–1200 times if our algorithms for time stepping, impact, etc. all are working well. This implies about $6\frac{1}{2}$ to 16 hours on the Cray for the job.

4. Shells

A substantial part of the modeling for which Abaqus is used involves shells, much of it in cases where geometric and material nonlinearities dominate the response. We have tried to provide useful capabilities for shell modeling, but we know that there are serious deficiencies in what we offer. Irons wrote often about shells: his definitive account of SemiLoof (Irons, 1976) and his textbook (Irons and Ahmad, 1980) both contain succinct statements of the difficulties. My impression is that things are improving rapidly. I hope so: I think we are all aware of the need.

In Abaqus we always treat shells as shells—we do not have any numerically degenerated solid elements acting as shells. The elements are formulated in terms of forces and moments at integration points. The behavior of the cross section is defined in closed form, or by numerical integration through the thickness. In most cases we use shear deformable elements, but the transverse shears are not considered in the constitutive calculations—we assume the shell is thin enough that transverse shear is not very important: it is simply a numerical technique that allows us to manage with low order interpolation. Low order elements are often desirable in practical cases.

Abaqus includes three types of shell model: axi-symmetric shells with axi-symmetric deformations, general shells, and pipes and elbows with deforming sections (ovalization and warping). These last elements are a rather special case which turns out to be important in certain piping applications that arise in the nuclear power industry and in the offshore oil industry. We often see problems where a mixture of shell and solid modeling is needed (K-joints in offshore platforms; Tees in pressure vessels). We have a standard technique for joining the shells to the solids, based on kinematic assumptions introduced as constraints. The approach appears to be satisfactory for the small strain cases where we have seen results.

The purely axi-symmetric case is rather simple because the problem is one dimensional. For this case we use a linear interpolation element with one integration point and a quadratic interpolation element with two. Our impression is that these elements are quite effective. Our implementation of these elements is based on a simplified finite membrane strain theory. The theory is similar in concept to Rodal's thesis (Rodal and Witmer, 1979), except that we use somewhat different strain measures because our applications involve material models for which logarithmic strain seems to be appropriate, and, in a case like this where the principal directions do not rotate, it is easy to work directly in terms of this strain. The main simplifying assumptions in the theory are that the thinning of the shell is uniform through the thickness and is defined by an incompressibility assumption on the reference surface of the shell, and that the thinning occurs smoothly, so that we can neglect gradients of thickness change with respect to position.

There are many applications for which a finite membrane strain model should be useful—obvious examples arise in sheet metal forming processes. The axi-symmetric case

is particularly simple and our version of a finite membrane strain theory is quite basic. Nevertheless, it takes two pages of the Abaqus theory manual just to write out the definition of the initial stress matrix, and the manipulations involved in reaching that definition are lengthy. The extension of the same formulation to general shell deformations will involve substantial algebraic manipulation.

The general shell elements that we use are the Batoz triangle, and the four and eight node "Ahmad quadrilaterals" (shear flexible elements using reduced integration). Our implementation of the Batoz triangle uses Batoz' interpolation functions in a co-rotational frame for bending, together with constant membrane strain. The element is in Abaqus because it is essential for us to offer a three-node triangular element for shell problems, and we have been advised that this element is among the better elements of this category. The element appears to behave well in bending—it converges rapidly and is relatively insensitive to distortion. The shortcomings of our implementation of the element are the constant membrane strain assumption, the faceted geometric approximation, and the need to use three integration points, this last because it makes the element more expensive than it should be, given the constant membrane strain assumption.

The Ahmad quadrilaterals are very attractive for cases involving material nonlinearity because the use of reduced integration minimizes the constitutive evaluations needed to form the element. The basis functions are also very simple. The four node element is of limited value without hourglass control: we use the hourglass controls proposed by Belytschko (Belytschko and Tsay, 1983) for this element. This improves the situation, but it is still not entirely satisfactory—we still find it necessary to change the hourglass stiffnesses from time to time, without being too sure of the reasons for the values we choose. Abaqus includes a print option which summarizes the energy content of the solution. It is not uncommon to see relatively significant "artificial strain energy" associated with the hourglass control: in typical high energy dynamic events this artificial strain energy may even exceed the residual "physical" energy (strain energy plus kinetic energy) at the end of the event. Yet the overall predictions of the significant aspects of the response are generally usable.

As Irons pointed out in his last paper (Irons and Loikkanen, 1983), the eight node Ahmad element does not pass the patch test except as a rectangle. We offer this element for historical reasons—it seemed to be a good choice at the time. In fact it does respectably well on shells that are not too thin, even if it is somewhat distorted. But there is another practical objection to the eight node interpolation scheme: contact problems are awkward with such functions. Contact over the entire element does not present serious difficulties, but, with the contact algorithms that we use (which have Lagrange multipliers to represent the contact pressure), we do not know how to deal with partial contact over an element—that is, with the possibility that contact or separation may occur over part of the surface represented by an element. We plan to add the nine-node version of the same element to overcome this last objection, even though Irons has told us that this form of the element only passes the patch test when it is geometrically bilinear (which would soon not be the case in a large displacement analysis, even if it were true to start with).

Both of the Ahmad elements, as we have implemented them, perform poorly if they are distorted and the shell is thin. This is a serious objection, but it would appear that

the element of K. C. Park (Park and Stanley, 1984) alleviates this problem. I think it is important, for practical cases, that we retain the simplicity of the low order elements as well as the cost effectiveness of reduced integration, especially when relatively severe nonlinearities (requiring, for example, modeling of finite strain effects) are present.

5. Integration of Conventional Plasticity Models

Conventional plasticity models present an interesting integration problem: the material behavior changes drastically at yield; the plastic strain is defined only as a rate, and the stress measures usually used are defined on the current configuration, so that the rotation of the material must be accounted for in some way. The problem has been receiving attention recently (for example Ortiz and Popov, 1984), and new approaches have been suggested. I would like to describe the approach that we have been using in Abaqus: overall, we are reasonably satisfied with it, but it raises some questions that we have not been able to answer, and which may also be relevant to the more recently proposed approaches.

Abaqus is primarily an implicit code, so that it is desirable to use an algorithm that is unconditionally stable. This is not strictly essential. In typical cases involving metals under ordinary conditions, our experience has been that we are unlikely to succeed in using increments in which the rotations exceed 10–15 degrees, or the strains exceed a few percent over important parts of the mesh (these limitations are associated with our Newton approach to solving the equations). Thus, any method that is stable for this size of increment should be satisfactory. However, I would be surprised to see conditionally stable algorithms that can handle such increments: typical yield strains in metals are about 10^{-3} , so that the increment size is about ten times the size of the yield surface in strain space, and I assume that the stability limit of a conditionally stable operator would not be larger than the yield strain. Because we have found the full Newton method to be effective, we want an algorithm that is sufficiently simple so that the Jacobian matrix, $(\partial \mathbf{T} / \partial \mathbf{E})_{t+\Delta t}$, where \mathbf{T} is the stress, \mathbf{E} is the strain, and $t + \Delta t$ is the time at the end of the increment (where we write the equilibrium equations in an implicit code), can be obtained exactly for common plasticity models. It is desirable that this matrix have an additional property: several plasticity models of practical importance are derived from a locally smooth potential which is also the yield surface, so that the rate equations of the model give a symmetric form $\partial \mathbf{T} / \partial \mathbf{E}$. We would like the integration operator to preserve this symmetry—this is of practical significance with respect to computational cost. The operator we use in Abaqus does not do so, except in a restricted class of plasticity models (which does not include some of those that are available in the code and which have a symmetric rate form).

We first integrate the kinematic aspect of the model by using the algorithm of Hughes and Winget (1980) to define an increment of rotation, which we apply to all vector and tensor valued functions at the material point; we also follow the Hughes-Winget suggestion and define the strain increment by the central difference approximation. Various authors have commented on the inadequacies of this class of method in the context of theories that involve tensor valued hardening variables, such as classical kinematic hardening theory. I do not see much utility in simple kinematic hardening with respect to finite strain applications, and the approach does not suffer the same obvious deficiency when it is used with

the isotropic yield models that we generally use for metals and soils at finite strains. The algorithm leads to a symmetric initial stress matrix, which is tedious to derive.

We are left with the integration of the change of state associated with deformation. The following discussion considers only isothermal, rate independent behavior with isotropic hardening. In Abaqus we use the same approach for rate dependent models (except where we integrate explicitly, as mentioned in Section 2.2 above), for non-isothermal cases (including fully coupled temperature-displacement calculations) and for more complicated hardening models, such as kinematic hardening.

The isotropic hardening plasticity models in Abaqus all have the following structure. We assume a strain rate decomposition,

$$d\mathbf{E} = d\mathbf{E}^{el} + d\mathbf{E}^{pl}$$

where $d\mathbf{E}$ is a differential change in total mechanical strain, $d\mathbf{E}^{el}$ is a differential change in the elastic strain, and $d\mathbf{E}^{pl}$ is a differential change in the inelastic (plastic) strain. We have introduced the Hughes-Winget approach to account, in an approximate way, for the rigid-body rotation of the material during the increment, and to define a finite increment of total strain during the increment. This allows us to write this decomposition in integrated form as

$$\mathbf{E} = \mathbf{E}^{el} + \mathbf{E}^{pl} \quad (1)$$

where \mathbf{E} , \mathbf{E}^{el} and \mathbf{E}^{pl} are defined as summations of the corresponding rotated values at the start of each increment and the incremental values associated with that increment. During the constitutive calculations \mathbf{E} is known, except in the case of plane stress.

We assume an elastic strain energy potential so that the stress, \mathbf{T} , is defined as

$$\mathbf{T} = \frac{\partial W}{\partial \mathbf{E}^{el}} \quad (2)$$

where $W = W(\mathbf{E}^{el}, \theta)$ is the strain energy potential, and θ is the temperature.

Some of the plasticity models that we use in Abaqus assume linear elasticity, while others (soils models) use a nonlinear elasticity in which the logarithm of the pressure stress is proportional to the volumetric strain. None of these elasticity models has internal constraints such as incompressibility, so that (2) defines the stress completely. It can be differentiated to give

$$\partial \mathbf{T} = \left[\frac{\partial^2 W}{\partial \mathbf{E}^{el} \partial \mathbf{E}^{el}} \right] : \partial \mathbf{E}^{el}$$

which we write as

$$\partial \mathbf{T} = [\mathbf{D}]^{el} : \partial \mathbf{E}^{el} \quad (3)$$

A simple isotropic hardening model has the yield function

$$f = 0 \quad (4)$$

where $f = f(\mathbf{T}, \theta, H^\alpha)$, with H^α being a set of hardening parameters; f is defined such that, whenever $f < 0$, the response is purely elastic. The models we use in Abaqus all have a smooth yield function, so that $\partial f / \partial \mathbf{T}$, $\partial^2 f / \partial \mathbf{T} \partial \mathbf{T}$, and $\partial f / \partial H^\alpha$ are well defined everywhere on f .

The flow rule is written

$$d\mathbf{E}^{pl} = d\lambda \frac{\partial g}{\partial \mathbf{T}} \quad (5)$$

where $d\lambda$ is a positive scalar and $g(\mathbf{T}, \theta, H^\alpha)$ is the flow potential. Since the material models being discussed are rate independent, $d\lambda$ is determined only by the kinematic solution at the material point being considered.

The hardening parameters evolve with plastic strain:

$$dH^\alpha = h^\alpha(d\mathbf{E}^{pl}, \mathbf{T}, \theta, H^\beta)$$

where h^α defines the hardening: h^α must be homogeneous of degree one in $d\mathbf{E}^{pl}$ for the model to be rate independent. Therefore

$$dH^\alpha = d\lambda h^\alpha\left(\frac{\partial g}{\partial \mathbf{T}}, \mathbf{T}, \theta, H^\beta\right) \quad (6)$$

The plasticity model is now defined, except for choosing particular forms for the elastic strain energy potential, W , the yield function, f , the flow potential, g , and the hardening rules, h^α .

The only rate equations in the formal definition of the material model are the evolutionary rule for the hardening and the flow rule. The simplest operator which may at least fulfill the requirement of unconditional stability mentioned at the beginning of this section is the backward Euler method, which can be introduced into (5) to give

$$\Delta \mathbf{E}^{pl} = \Delta \lambda \frac{\partial g}{\partial \mathbf{T}} \quad (7)$$

while (6) is written

$$\Delta H^\alpha = \Delta \lambda h^\alpha\left(\frac{\partial g}{\partial \mathbf{T}}, \mathbf{T}, \theta, H^\beta\right) \quad (8)$$

In these equations and in all of the following, all quantities are evaluated at the end of the time increment.

Remark: Ortiz and Popov (1984) propose the midpoint rule as a more accurate operator, based on an error analysis with small strain increments. Their analysis appears to assume that the material is yielding during the entire increment. This is often not the case—in most calculations there are always material points where yielding begins part way through the increment. At such points the mid-point rule requires solution for the initial intersection with the yield surface during the increment, and application of the rule only within the

yielding part of the increment. This creates some formidable algebra, especially if one wishes to derive the Jacobian. Our analysis of this problem leads us to the conclusion that, even for an associated flow material, the Jacobian will, in this case, not be symmetric, and that its non-symmetric part may not be very small. We choose the backward Euler method partly because it does not introduce this difficulty, and so retains sufficient tractability that we can complete the algebra rather easily. It is interesting to note that Ortiz and Popov confirm the conclusion of Schreyer et al. (1979): that, for strain increments that are not small compared to the size of the yield surface in strain space, the backward Euler method exhibits the highest accuracy of any of the simple methods that they tested.

From a computational viewpoint the problem is now algebraic: we must solve the nonlinear equations (1), (2), (4), (7) and (8), and thus define the state of the material at the end of the increment. The “material stiffness matrix”,

$$[\mathbf{D}] = \left[\frac{\partial \mathbf{T}}{\partial \mathbf{E}} \right]$$

must also be defined for the overall Newton scheme that we use for the equilibrium equations. Simple manipulations lead to the definition

$$[\mathbf{D}] = [[\mathbf{I}] + \Delta\lambda[\mathbf{H}] : [\mathbf{L}]]^{-1}[\mathbf{H}]$$

where $[\mathbf{I}]$ is the fourth order identity tensor, and

$$[\mathbf{H}] = [\mathbf{D}]^{el} : [\mathbf{Z}]$$

$$[\mathbf{Z}] = [\mathbf{I}] - \frac{1}{d}\hat{\mathbf{N}}\mathbf{M} : [\mathbf{D}]^{el}$$

$$\hat{\mathbf{N}} = \mathbf{N} + \Delta\lambda \frac{\partial \mathbf{N}}{\partial H^\alpha} c_{\alpha\beta} h^\beta$$

$$\mathbf{N} = \frac{\partial g}{\partial \mathbf{T}}$$

$$c_{\alpha\beta} = \left(\delta_{\alpha\beta} - \Delta\lambda \left(\frac{\partial h^\alpha}{\mathbf{N}} : \frac{\partial \mathbf{N}}{\partial H^\beta} + \frac{\partial h^\alpha}{\partial H^\beta} \right) \right)^{-1}$$

$$d = \mathbf{M} : [\mathbf{D}]^{el} : \hat{\mathbf{N}} - \frac{\partial f}{\partial H^\alpha} c_{\alpha\beta} h^\beta$$

$$\mathbf{M} = \frac{\partial f}{\partial \mathbf{T}} + \Delta\lambda \frac{\partial f}{\partial H^\alpha} c_{\alpha\beta} \left(\frac{\partial h^\beta}{\partial \mathbf{N}} : \left[\frac{\partial \mathbf{N}}{\partial \mathbf{T}} \right] + \frac{\partial h^\beta}{\partial \mathbf{T}} \right)$$

and

$$[\mathbf{L}] = \frac{\partial \mathbf{N}}{\partial \mathbf{T}} + \Delta\lambda \left(\left[\frac{\partial \mathbf{N}}{\partial H^\alpha} c_{\alpha\beta} \frac{\partial h^\beta}{\partial \mathbf{N}} \right] : \left[\frac{\partial \mathbf{N}}{\partial \mathbf{T}} \right] + \frac{\partial \mathbf{N}}{\partial H^\alpha} \frac{\partial h^\alpha}{\partial \mathbf{T}} \right)$$

For a non-associated flow material, $\partial f / \partial \mathbf{T}$ and \mathbf{N} are not related, so that $[\mathbf{D}]$ cannot be symmetric. For an associated flow material, $\partial f / \partial \mathbf{T}$ and \mathbf{N} (and hence $\hat{\mathbf{N}}$) will be co-linear. Therefore $[\mathbf{D}]$ will be symmetric if

$$\frac{\partial h^\alpha}{\partial \mathbf{N}} : \left[\frac{\partial \mathbf{N}}{\partial \mathbf{T}} \right] = 0 \quad \text{and} \quad \frac{\partial h^\alpha}{\partial \mathbf{T}} = 0$$

which is the case, for example, for the simple Prandtl-Reuss model that is commonly used for metals, but is not true for the modified Cam-clay model that is sometimes used for clays. However, in associated flow cases when it is not true, the non-symmetric contributions to $[\mathbf{D}]$ appear only in the terms multiplied by $(\Delta\lambda)^2$, so that they are of the order of (the plastic strain increment)² compared to unity. This suggests that, for practical purposes, the lack of symmetry should not degrade the convergence of the Newton iterations for equilibrium if we approximate $[\mathbf{D}]$ with its symmetric part, and our experience to date confirms this—at least, the performance of the algorithm with the symmetric approximation has been satisfactory, by our standards.

There remains the problem of solving the algebraic equations for the state at the end of the increment. In the general case, this is not a simple matter: whatever method is used should not be expensive on the computer, but it must work for the tightly curved yield functions that appear in some models of practical interest, even when the strain increment is many times the typical size of the yield surface in strain space. The problem is made more awkward in cases such as Cam-clay, where exponential terms appear in the elasticity and in the hardening. Our approach has been as follows. A Newton scheme should work, provided we choose suitable variables as a basis, and provided we start with a good guess. The plastic strain increments, $\Delta \mathbf{E}^{pl}$, should be a suitable set of variables. Then Newton's method for (1), (2), (4), (7) and (8) is the linear equations

$$([\mathbf{I}] + \Delta\lambda[\mathbf{Z}] : [\mathbf{L}] : [\mathbf{D}]^{el}) : \mathbf{C}^{pl} = [\mathbf{Z}] : (\Delta\lambda \mathbf{N} - \Delta \mathbf{E}^{pl}) + \hat{\mathbf{N}} \frac{1}{d} f$$

which are solved for \mathbf{C}^{pl} , the improvement to the solution for $\Delta \mathbf{E}^{pl}$:

$$\Delta \mathbf{E}^{pl} = \Delta \mathbf{E}^{pl} + \mathbf{C}^{pl}$$

The elastic strain is then obtained from (1), the stress from (2). $\Delta\lambda$ from the projection of (7) onto \mathbf{N} :

$$\Delta\lambda = \frac{\mathbf{N} : \Delta \mathbf{E}^{pl}}{\mathbf{N} : \mathbf{N}}$$

and ΔH^α from (8).

This loop is repeated until the components of $\Delta \mathbf{E}^{pl}$ converge. A tight convergence criterion is necessary, so that the solution is accurately defined: this is essential for the overall Newton scheme for the equilibrium equations to converge quadratically.

The Newton method requires solution of the linear system at each iteration. Although the system of equations is not large (at most six), these computations are done at each integration point and at each iteration of the overall equilibrium solution, so that it is desirable to solve the problem more efficiently than by direct application of the Newton method. In addition, when the strain increment is large, a reasonable starting guess is necessary for the method to converge. For these reasons we have been using a projection of the problem onto a smaller number of variables to start the solution. If this subspace is chosen appropriately, we should be able to develop a useful estimate of the solution at low cost. The general idea is as follows.

Let $\tilde{\mathbf{K}}^\alpha$, $\alpha = 1, \dots, n$, be a set of tensors that are orthonormal:

$$\tilde{\mathbf{K}}^\alpha : \tilde{\mathbf{K}}^\beta = \delta^{\alpha\beta}$$

and choose n to be less than the number of stress components in the actual problem.

The $\tilde{\mathbf{K}}^\alpha$ are chosen for a particular guess and are fixed during the solution for that guess.

Assume that the plastic strain increment is

$$\Delta \mathbf{E}^{pl} = \Delta e_\alpha^{pl} \tilde{\mathbf{K}}^\alpha \quad (9)$$

The elasticity, (2), and the strain rate decomposition, (1), then define the stress, which we require to satisfy yield, (4). The integrated flow equation, (7), is imposed in the sub-space:

$$\Delta \mathbf{E}^{pl} = \Delta \tilde{\lambda} \tilde{\mathbf{N}}$$

where

$$\tilde{\mathbf{N}} = \tilde{\mathbf{K}}^\alpha \tilde{\mathbf{K}}^\alpha : \mathbf{N} \quad (10)$$

Since the $\tilde{\mathbf{K}}^\alpha$ are orthonormal, (9) and (10) then define

$$\Delta e_\alpha^{pl} = \Delta \tilde{\lambda} N^\alpha \quad (11)$$

where

$$N^\alpha = \tilde{\mathbf{K}}^\alpha : \mathbf{N}$$

The Newton scheme described above for the full stress space projects directly:

$$[\delta^{\alpha\beta} + \Delta \tilde{\lambda} Z^{\alpha\gamma} (\hat{\mathbf{K}}^\gamma : [\mathbf{L}] : [\mathbf{D}]^{el} : \hat{\mathbf{K}}^\beta)] c^\beta = Z^{\alpha\beta} (\Delta \tilde{\lambda} N^\beta - \Delta e_\beta^{pl}) + \hat{N}^\alpha \frac{f}{\tilde{d}}$$

where

$$Z^{\alpha\beta} = \delta^{\alpha\beta} - \frac{1}{\tilde{d}} \tilde{\mathbf{K}}^\alpha : \hat{\mathbf{N}} \mathbf{M} : [\mathbf{D}]^{el} : \tilde{\mathbf{K}}^\beta$$

$$\hat{N}^\alpha = \tilde{\mathbf{K}}^\alpha : \hat{\mathbf{N}}$$

and

$$\tilde{d} = \mathbf{M} : [\mathbf{D}]^{el} : \tilde{\mathbf{K}}^\alpha \tilde{\mathbf{K}}^\alpha : \hat{\mathbf{N}} - \frac{\partial f}{\partial H_\alpha} c_{\alpha\beta} h^\beta$$

These linear equations provide the c^α , the improvements to the Δe_α^{pl} :

$$\Delta e_\alpha^{pl} = \Delta e_\alpha^{pl} + c^\alpha$$

At each iteration the solution is updated as described for the full stress-space solution, except that $\Delta \tilde{\lambda}$ is calculated by projecting (11) onto N^α :

$$\Delta \tilde{\lambda} = \frac{N^\alpha \Delta e_\alpha^{pl}}{N^\beta N^\beta}$$

The utility of this technique depends on the choice of the $\tilde{\mathbf{K}}^\alpha$, and on n . Clearly it is unlikely that it would be worthwhile to use $n > 2$. Most of the plasticity models in Abaqus are isotropic, in the sense that the yield function, f , and the flow potential, g , are defined in terms of the stress invariants. An obvious choice for the $\tilde{\mathbf{K}}^\alpha$ in these cases is $n = 2$, with

$$\tilde{\mathbf{K}}^1 = \frac{1}{\sqrt{3}} \mathbf{I}$$

(here \mathbf{I} is the identity matrix),

and

$$\tilde{\mathbf{K}}^2 = \sqrt{\frac{3}{2}} \frac{\mathbf{S}^0}{q^0}$$

where \mathbf{S}^0 is the deviatoric part of \mathbf{T}^0 , the stress that would occur at the end of the increment if there were no plasticity occurring in the increment, and

$$q^0 = \sqrt{\frac{3}{2} \mathbf{S}^0 : \mathbf{S}^0}$$

If the only stress dependencies in f and g are the effective pressure stress,

$$p = -\frac{1}{3} \mathbf{I} : \mathbf{T}$$

and the deviatoric stress magnitude, q ; and the elasticity is isotropic, (and the hardening is isotropic, as has been assumed here), the subspace solution is the exact answer to the

problem, except for the plane stress case. For some simple yield surfaces and hardening definitions the subspace problem can be solved in closed form, without the need for iteration. For the simplest case of a Mises material, where q is the only stress term in f and g , the one-dimensional sub-space defined by $\tilde{\mathbf{K}}^2$ above provides the exact solution. For perfect plasticity the method is then precisely Wilkins' "radial return" algorithm.

Our experience with more general yield functions and flow potentials, where the third stress invariant is also used, is that the two-dimensional sub-space provides a satisfactory guess, from which the solution can be completed usually with no more than two iterations of the full stress-space problem.

6. Closure

The paper has presented a brief review of our experience in the recent past in the general area of nonlinear structural analysis of metal shells. It is not difficult to identify the most severe limitations in the methods we have used so far, to write down a "wish list" of areas where we would like to be using better methods than those we have used to date. The list is as follows.

- We do not like the rapid deterioration in the accuracy of the isoparametric shell elements that we currently use when they are not rectangles and the shell is thin.
- Reduced integration elements are attractive because they minimize constitutive calculations, which are often a significant part of the computational cost when the material model is not simple. Numerical artifices, such as hourglass control, are not objectionable, when they are effective and are well understood. Our own understanding so far is lacking.
- The characteristic hinging of thin metal shells under compressive loads raises difficulties and opportunities. The difficulties are associated with concerns about capturing this behavior with smooth discretizations. The opportunities are indicated by the remarkable success of the rigid-plastic models of Wierzbicki. Perhaps it is true that, in practical cases, load bearing members are too thick for this to be important, although movies of front-end collision experiments on cars shown very pronounced hinging.
- A straightforward finite membrane strain shell formulation would have wide application in several problem areas that we often encounter. The axi-symmetric formulation we have been using is based substantially on the assumptions introduced by Rodal in his thesis, although we have taken a rather different approach in detail. We do not have a good appreciation of the limitations of the formulation.
- We expect to continue to work with implicit methods in many problem areas, including a large part of the analysis work associated with structural design. If this is a correct assessment, it would be highly desirable to have equation solution methods that are more effective and less difficult to use than Newton's method. This is, from our point of view, one of the most severe limitations that we face in practical applications.
- So much shell behavior involves branching on the solution path that it would be very satisfying to have methods that can handle this unassisted.

- Finally, we have not mentioned rezoning, or the treatment of localization, but these are becoming important issues in practical cases.

7. References

T. Belytschko and C.-S. Tsay, "A Stabilization Procedure for the Quadrilateral Plate Element with One-Point Quadrature", *Int. J. Num. Meth. Eng.*, Vol. 19, pp. 405-419, 1983.

H.M. Hilber, T.J.R. Hughes and R.L. Taylor, "Collocation, Dissipation and Overshoot for Time Integration Schemes in Structural Dynamics", *Earthquake Eng. and Struc. Dyn.*, Vol. 16, pp. 99-117, 1978.

T. J. R. Hughes and J. Winget, "Finite Rotation Effects in Numerical Integration of Rate Constitutive Equations Arising in Large Deformation Analysis", *Int. J. Num. Meth. Eng.*, Vol. 15, pp. 1862-1867, 1980.

B. M. Irons, "The Semi-Loof Shell Elements", Chapter 11 of *Finite Elements for Thin Shells & Curved Members*, edited by D. G. Ashwell and R. H. Gallagher, John Wiley & Sons, London, 1976.

B. M. Irons and S. Ahmad, *Techniques of Finite Elements*, Ellis Horwood Ltd., Chichester, England, 1980.

B. M. Irons and M. Loikkanen, "An Engineers' Defense of the Patch Test", *Int. J. Num. Meth. Eng.*, Vol. 19, pp. 1391-1401, 1983.

M. Ortiz and E. P. Popov, "Accuracy and Stability of Integration Algorithms for Elastoplastic Constitutive Relations", Brown University, Division of Engineering Report, 1984.

K. C. Park and G. M. Stanley, "A Curved C^0 Shell Element Based On Assumed Natural-Coordinate Strains", Applied Mechanics Lab., Lockheed Palo Alto Research Lab., Palo Alto, Cal., 1984.

E. Riks, "An Incremental Approach to the Solution of Snapping and Buckling Problems", *Int. J. Solids and Struc.*, Vol. 15, pp. 329-551, 1979.

J. J. A. Rodal and E. A. Witmer, "Finite-Strain Large-Deflection Elastic-Viscoplastic Finite-Element Transient Response Analysis of Structures", contractor report 159874, NASA Lewis, 1979.

J. C. Simo and M. Ortiz, "A Unified Approach to Finite Deformation Elastoplastic Analysis Based on the Use of Hyperelastic Constitutive Equations", Brown University Division of Engineering Report, 1984.

G. Strang, and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, New Jersey, 1973.

H. L. Schreyer, R. F. Kulak and J. M. Kramer, "Accurate Numerical Solutions for Elastic-Plastic Models", *Trans. ASME, J. Pressure Vessel Technology*, Vol. 101, 1979.

P. S. Symonds, "Survey of Methods of Analysis for Plastic Deformation of Structures under Dynamic Loading", Brown University Division of Engineering Report, 1967.